ASSIGNMENT

**NAME:** D.Harshitha

Code:

import numpy as np

import pandas as pd

from scipy import stats

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style("darkgrid")

train = pd.read\_csv("/content/train.csv")

test = pd.read\_csv("/content/test.csv")

print("train shape : ", train.shape)

print("test shape : ", test.shape)

train.head()

train.describe()

test.describe()

train["Survived"].value\_counts().plot(kind="pie",explode=[0,0.05],autopct='%1.1f%%')

plt.title("Ratio of Survived",weight = "bold")

train1 = train[train.columns[train.isnull().sum()!=0]]

test1 = test[test.columns[test.isnull().sum()!=0]]

na\_prop1 = (train1.isnull().sum()).sort\_values()

na\_prop2 = (test1.isnull().sum()).sort\_values()

plt.figure(figsize=(10,8))

sns.set\_style('whitegrid')

plt.subplot(211)

na\_prop1.plot.barh(color='blue')

plt.title('Missing values(train set)', weight='bold')

plt.subplot(212)

na\_prop2.plot.barh(color='blue')

plt.title('Missing values(test set)', weight='bold' )

train1 = train.select\_dtypes(include="object").drop(["Name", "Ticket", "Cabin"], axis=1)

plt.figure(figsize=(12,6))

plt.style.use('ggplot')

for i, col in enumerate(train1.columns):

    plt.subplot(1, 2, i+1)

    sns.countplot(x=col, hue="Survived", data=train)

    plt.title(f'Survival Count by {col}')

train1 = train.select\_dtypes(include="int").drop(["PassengerId", "Survived"], axis=1)

plt.figure(figsize=(12,5))

plt.style.use('ggplot')

for i, col in enumerate(train1.columns):

    plt.subplot(1, 3, i+1)

    sns.countplot(x=col, hue="Survived", data=train)

    plt.title(f'Survival by {col}')

train1 = train.select\_dtypes(include = "float")

plt.figure(figsize=(12,5))

plt.style.use('ggplot')

for i, col in enumerate(train1.columns):

    plt.subplot(1,2,i+1)

    sns.distplot(train[train["Survived"]==0][col],color='red')

    sns.distplot(train[train["Survived"]==1][col],color='blue')

plt.figure(figsize=(12,5))

train["Family"] = train["SibSp"] + train["Parch"] + 1

sns.countplot(x="Family", hue="Survived", data=train)

plt.title("Survival Count by Family Size")

train['Age\_group'] = pd.qcut(train['Age'], 10)

plt.figure(figsize=(14,7))

sns.countplot(x='Age\_group', hue='Survived', data=tra

plt.figure(figsize=(14,5))

train["Cabin"] = train["Cabin"].fillna("None")

train["Cabin\_First"] = train["Cabin"].apply(lambda x: x[0] if x != "None" else "N")

plt.subplot(121)

sns.barplot(x="Cabin\_First", y="Survived", data=train)

plt.title("Cabin First vs Survived")

plt.subplot(122)

sns.countplot(x="Cabin\_First", data=train[train["Cabin\_First"] != "N"])

plt.title("Cabin First Count (without NA value)")

cabin\_pcl\_group = train.groupby(["Cabin\_First","Pclass"])["PassengerId"].count()

cabin\_pcl\_pers = {}

for i in train["Cabin\_First"].unique():

    cabin\_pcl\_pers[i] = cabin\_pcl\_group[i] / cabin\_pcl\_group[i].sum()

    for pcl in range(1,4):

        try:

            cabin\_pcl\_pers[i][pcl]

        except:

            cabin\_pcl\_pers[i][pcl] = 0

cabin\_pcl\_pers = pd.DataFrame(cabin\_pcl\_pers).T

plt.figure(figsize = (11,5))

plt.bar(np.arange(9),cabin\_pcl\_pers[1], color='skyblue', width=0.7, label='Pclass = 1')

plt.bar(np.arange(9),cabin\_pcl\_pers[2], bottom=cabin\_pcl\_pers[1], color='lightpink', width=0.7, label='Pclass = 2')

plt.bar(np.arange(9),cabin\_pcl\_pers[3], bottom=cabin\_pcl\_pers[1] + cabin\_pcl\_pers[2], color='lightgray', width=0.7, label='Pclass = 3')

plt.xticks(np.arange(9),cabin\_pcl\_pers.index)

plt.legend(loc='upper left', bbox\_to\_anchor=(1, 1))

plt.title('Percentage of Pclass with Cabin')

train['Name'].head()

train['First\_Name\_label'] = [0 if name=='Mr' else 1 if name=='Mrs'  else 2 if name=='Miss' else 3 if name=='Master' else 4 for name in train['First\_Name']]

plt.figure(figsize=(8,5))

sns.barplot(x='First\_Name\_label', y='Survived', data=train)

plt.xticks(np.arange(5),['Mr','Mrs','Miss','Master','Other'])

print(train["Ticket"].head(10))

print("Ticket unique value number : " , train["Ticket"].nunique())

ticket\_group = train.groupby('Ticket')['PassengerId'].count().reset\_index()

ticket\_group.columns = ['Ticket',"Ticket\_freq"]

train\_t = train.merge(ticket\_group, on = "Ticket")

plt.figure(figsize=(13,10))

plt.subplot(211)

sns.countplot(x='Ticket\_freq', hue='Survived', data=train\_t)

plt.subplot(212)

sns.barplot(x='Ticket\_freq', y='Survived', data=train\_t)

train\_group = train.groupby(['Pclass','Sex'])['Survived'].mean()

plt.figure(figsize=(10,7)) train.groupby(['Sex','Pclass','Embarked'])['Survived'].mean()

import seaborn as snsfrom scipy.stats import chi2\_contingency

df1 = train.groupby(["Embarked","Survived"])["Survived"].count().unstack("Survived")

chi2, p, dof, expected = chi2\_contingency(df1)

msg = 'Embarked Statistic: {}\np-value: {}\nDegree of Freedom: {}'

print(msg.format(chi2, p, dof))

df2 = train.groupby(["Pclass","Survived"])["Survived"].count().unstack("Survived")

chi2, p, dof, expected = chi2\_contingency(df2)

msg = 'Pclass Statistic: {}\np-value: {}\nDegree of Freedom: {}'

print(msg.format(chi2, p, dof))

df3 = train.groupby(["Sex","Survived"])["Survived"].count().unstack("Survived")

chi2, p, dof, expected = chi2\_contingency(df3)

msg = 'Sex Statistic: {}\np-value: {}\nDegree of Freedom: {}'

print(msg.format(chi2, p, dof))

sns.catplot(x="Embarked", y="Survived", hue="Pclass", data=train, kind="bar")

sns.heatmap(train\_group.unstack("Pclass"), cmap='Blues', annot=True)

train\_group = train.groupby(['Pclass','Embarked'])['Survived'].mean()

plt.figure(figsize=(10,7))

sns.heatmap(train\_group.unstack("Pclass"), cmap='Blues', annot=True)

plt.figure(figsize=(8, 8))

numeric\_df = train.select\_dtypes(include=["int64", "float64"])

numeric\_df = numeric\_df.drop("PassengerId", axis=1)

sns.heatmap(numeric\_df.corr(), cmap="Blues", annot=True)

plt.title("Correlation Heatmap")

plt.show()

output:









